06 - List in Python

Ex. No. : 6.1 **Date: 4/05/2024**

Register No.: 231401001 **Name: Aafrin Fathima N**

Balanced Array

Given an array of numbers, find the index of the smallest array element (the pivot), for which the sums of all elements to the left and to the right are equal. The array may not be reordered.

Example
arr=[1,2,3,4,6]

- the sum of the first three elements, 1+2+3=6. The value of the last element is 6.
- Using zero based indexing, arr[3]=4 is the pivot between the two subarrays.
- The index of the pivot is 3.

Constraints

- $3 \le n \le 10^5$
- $1 \le arr[i] \le 2 \times 10^4$, where $0 \le i < n$
- It is guaranteed that a solution always exists.

The first line contains an integer n, the size of the array arr.
Each of the next n lines contains an integer, arr[i], where $0 \le i < n$.

Sample Case 0
Sample Input 0
4
1
2
3
3
Sample Output 0
2

Explanation 0

- The sum of the first two elements, 1+2=3. The value of the last element is 3.
- Using zero based indexing, arr[2]=3 is the pivot between the two subarrays.
- The index of the pivot is 2.

Sample Case 1
Sample Input 1
3
1
2
1
Sample Output 1

1

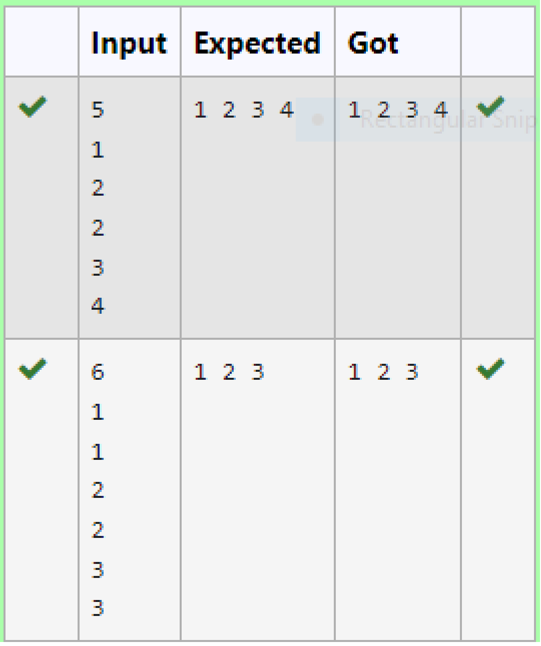
Explanation 1

- The first and last elements are equal to 1.
- Using zero based indexing, arr[1]=2 is the pivot between the two subarrays.
- The index of the pivot is 1.

For example:

Input	Result
4 1 2 3 3	2
3 1 2 1	1

```
n = int(input())
arr = [int(input()) for _ in range(n)]
n = len(arr)
total_sum = sum(arr)
left_sum = 0
flag=0
for i in range(n):
    if left_sum == total_sum - arr[i] - left_sum:
        print(i)
        flag=1
    left_sum += arr[i]
if flag==0:
    print("-1")
```



	Input	Expected	Got	
✓	5 1 2 2 3 4	1 2 3 4	1 2 3 4	✓
✓	6 1 1 2 2 3 3	1 2 3	1 2 3	✓

Ex. No. : 6.2 **Date: 4/05/2024**

Register No.: 231401001 **Name: Aafrin Fathima N**

Check pair with difference k

Given an array A of sorted integers and another non negative integer k, find if there exists 2 indices i and j such that A[i] - A[j] = k, i != j.

Input Format

1. First line is number of test cases T. Following T lines contain:
2. N, followed by N integers of the array
3. The non-negative integer k

Output format

Print 1 if such a pair exists and 0 if it doesn't.

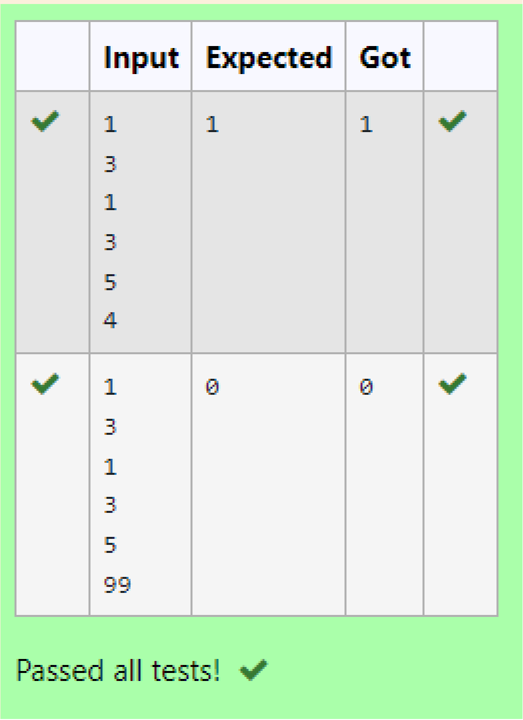
Input
1
3
1
3
5
4
Output:
1
Input
1
3
1
3
5
99
Output
0

For example:

Input	Result
1 3 1 3 5 4	1

Input	Result
1 3 1 3 5 99	0

```
T = int(input())
for _ in range(T):
    N = int(input())
    A = [int(input()) for _ in range(N)]
    k = int(input())
    has_pair = any(abs(a - b) == k for i, a in enumerate(A) for b in A[i + 1:])
    print("1" if has_pair else "0")
```



	Input	Expected	Got	
✓	1 3 1 3 5 4	1	1	✓
✓	1 3 1 3 5 99	0	0	✓

Passed all tests! ✓

Ex. No. : 6.3 **Date: 4/05/2024**

Register No.: 231401001 **Name: Aafrin Fathima N**

Count Elements

Complete the program to count frequency of each element of an array. Frequency of a particular element will be printed once.

Sample Test Cases

Test Case 1

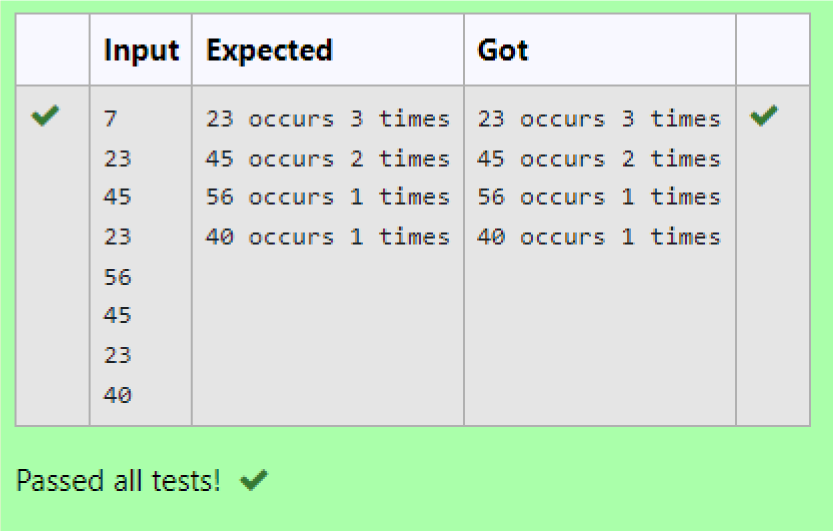
Input

```
 7
23
45
23
56
45
23
40
```

Output

```
 23 occurs 3 times
45 occurs 2 times
56 occurs 1 times
40 occurs 1 times
```

```
n=int(input())
arr = [int(input()) for _ in range(n)]
freq = {}
for i in arr:
    if i in freq:
        freq[i] += 1
    else:
        freq[i] = 1
for num, count in freq.items():
        print(num,"occurs",count,"times")
```



	Input	Expected	Got	
✓	7 23 45 23 56 45 23 40	23 occurs 3 times 45 occurs 2 times 56 occurs 1 times 40 occurs 1 times	23 occurs 3 times 45 occurs 2 times 56 occurs 1 times 40 occurs 1 times	✓

Passed all tests! ✓

Ex. No. : 6.4 **Date: 4/05/2024**

Register No.: 231401001 **Name: Aafrin Fathima N**

Distinct Elements in an Array

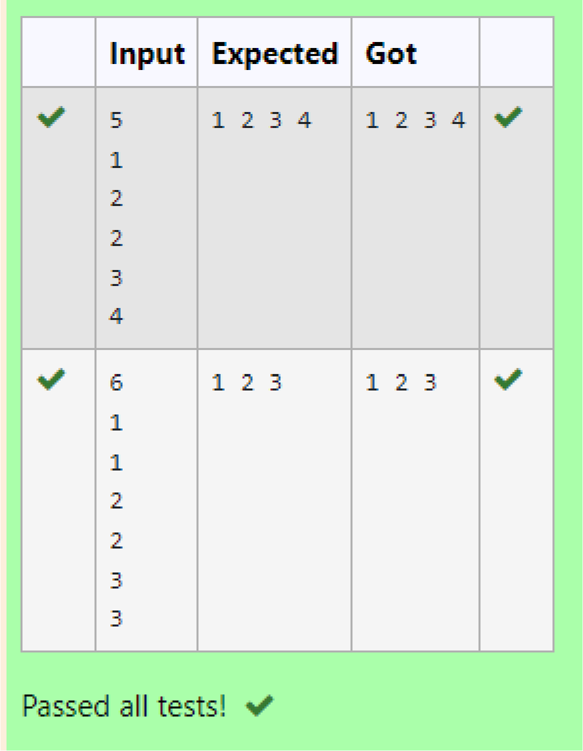
Program to print all the distinct elements in an array. Distinct elements are nothing but the unique (non-duplicate) elements present in the given array.
Input Format:
First line take an Integer input from stdin which is array length n.
Second line take n Integers which is inputs of array.
Output Format:
Print the Distinct Elements in Array in single line which is space Separated

Example Input:
5
1
2
2
3
4
Output:
1 2 3 4
Example Input:
6
1
1
2
2
3
3
Output:
1 2 3

For example:
Input Result
5
1
2
2
3
4
1 2 3 4
6

```
1
1
2
2
3
3
1 2 3
```

```
N1 = int(input())

array1 = {int(input()) for _ in range(N1)}

N2 = int(input())

array2 = {int(input()) for _ in range(N2)}

merged_array = array1 | array2

print(' '.join(map(str, sorted(merged_array))))
```



	Input	Expected	Got	
✓	5 1 2 2 3 4	1 2 3 4	1 2 3 4	✓
✓	6 1 1 2 2 3 3	1 2 3	1 2 3	✓

Passed all tests! ✓

Ex. No. : 6.5 **Date: 4/05/2024**

Register No.: 231401001 **Name: Aafrin Fathima N**

Element Insertion

Consider a program to insert an element / item in the sorted array. Complete the logic by filling up required code in editable section. Consider an array of size 10. The eleventh item is the data is to be inserted.

Sample Test Cases

Test Case 1

Input

```
1
3
4
5
6
7
8
9
10
11
2
```

Output

```
ITEM to be inserted:2
After insertion array is:
1
2
3
4
5
6
7
8
9
10
11
```

Test Case 2

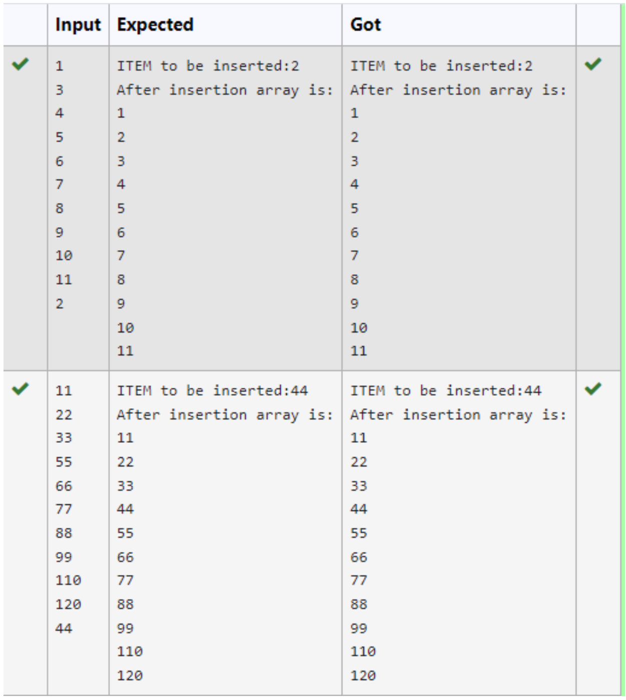
Input

```
11
22
33
55
66
77
88
99
110
120
44
```

Output

```

ITEM to be inserted:44
After insertion array is:
11
22
33
44
55
66
77
88
99
110
120
```

```
l=[]
for i in range(11):
    v=int(input())
    l.append(v)
print("ITEM to be inserted:",end='')
print(l[10])
print("After insertion array is:")
l.sort()
for i in l:
    print(i)
```



	Input	Expected	Got	
✓	1 3 4 5 6 7 8 9 10 11 2	ITEM to be inserted:2 After insertion array is: 1 2 3 4 5 6 7 8 9 10 11	ITEM to be inserted:2 After insertion array is: 1 2 3 4 5 6 7 8 9 10 11	✓
✓	11 22 33 55 66 77 88 99 110 120 44	ITEM to be inserted:44 After insertion array is: 11 22 33 44 55 66 77 88 99 110 120	ITEM to be inserted:44 After insertion array is: 11 22 33 44 55 66 77 88 99 110 120	✓

Ex. No. : 6.6 **Date: 4/05/2024**

Register No.: 231401001 **Name: Aafrin Fathima N**

Find the Factor

Determine the factors of a number (i.e., all positive integer values that evenly divide into a number) and then return the p^{th} element of the list, sorted ascending. If there is no p^{th} element, return 0.

Constraints

$1 \leq n \leq 10^{15}$

$1 \leq p \leq 10^{9}$

The first line contains an integer n, the number to factor.

The second line contains an integer p, the 1-based index of the factor to return.

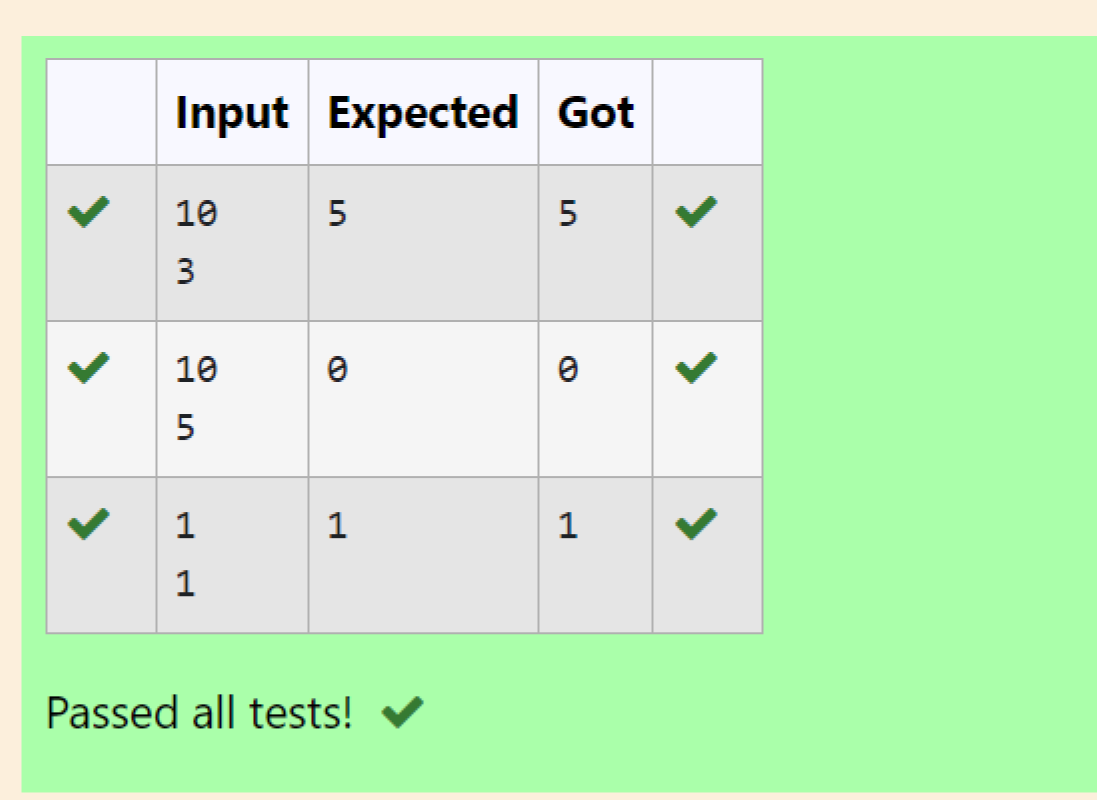
Sample Case 0
Sample Input 0
10
3
Sample Output 0
5
Explanation 0
Factoring n = 10 results in {1, 2, 5, 10}. Return the p = 3rd factor, 5, as the answer.
Sample Case 1
Sample Input 1
10
5
Sample Output 1
0
Explanation 1
Factoring n = 10 results in {1, 2, 5, 10}. There are only 4 factors and p = 5, therefore 0 is returned as the answer.
Sample Case 2
Sample Input 2
1
1
Sample Output 2
1
Explanation 2
Factoring n = 1 results in {1}. The p = 1st factor of 1 is returned as the answer.

For example:

Input	Result
10 3	5
10 5	0
1 1	1

Program:

```
def find_pth_factor(n, p):
  """
  Finds the pth factor (ascending order) of a number n, or 0 if it doesn't exist.

  Args:
      n: The number to factor.
      p: The 1-based index of the factor to return.

  Returns:
      The pth factor of n or 0 if it doesn't exist.
  """

  # Handle edge cases (n <= 1)
  if n <= 1:
    return 1 if p == 1 else 0  # Only 1 factor for n <= 1
  factors = []
  # Find factors up to the square root of n (efficient)
  for i in range(1, int(n**0.5) + 1):
    if n % i == 0:
      factors.append(i)
      # If i is not the square root, add its pair for complete factorization
      if i * i != n:
        factors.append(n // i)
# Sort factors in ascending order
  factors.sort()

  # Check if pth factor exists and return it or 0
  return factors[p - 1] if p <= len(factors) else 0


# Get input from user
n = int(input())
p = int(input())


# Find and print the pth factor
result = find_pth_factor(n, p)
print(result)
```



	Input	Expected	Got	
✓	10 3	5	5	✓
✓	10 5	0	0	✓
✓	1 1	1	1	✓

Passed all tests! ✓

Ex. No. : 6.7 **Date: 4/05/2024**

Register No.: 231401001 **Name: Aafrin Fathima N**

Merge List

Write a Python program to Zip two given lists of lists.

Input:
m : row size
n: column size
list1 and list 2 : Two lists
Output
Zipped List : List which combined both list1 and list2

Sample test case

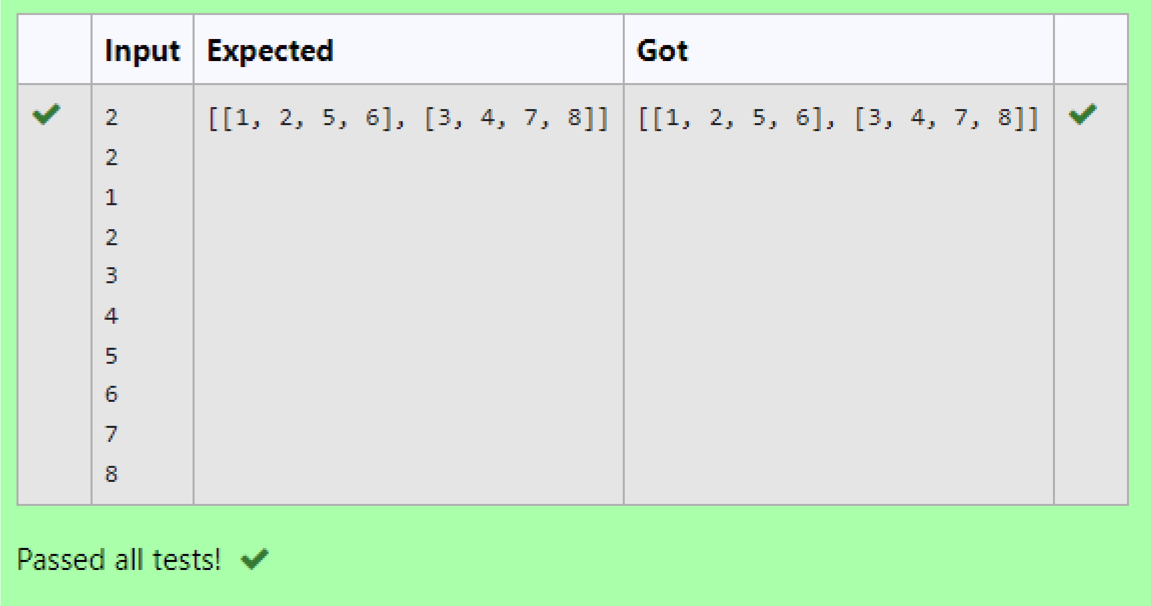
Sample input

```
2
2
1
3
5
7
2
4
6
8
```

Sample Output

```
[[1, 3, 2, 4], [5, 7, 6, 8]]
```

```
n=int(input())

m=int(input())

a=[]

b=[]

for i in range(2):

   for i in range(n):
        v=int(input())
        a+=[v,]
    for i in range(m):
        v=int(input())
        b+=[v,]
new=[a,b]
print(new)
```



	Input	Expected	Got	
✓	2 2 1 2 3 4 5 6 7 8	[[1, 2, 5, 6], [3, 4, 7, 8]]	[[1, 2, 5, 6], [3, 4, 7, 8]]	✓

Passed all tests! ✓

Ex. No. : 6.8 **Date: 4/05/2024**

Register No.: 231401001 **Name: Aafrin Fathima N**

Merge Two Sorted Arrays Without Duplication

Output is a merged array without duplicates.

Input Format
N1 - no of elements in array 1
Array elements for array 1
N2 - no of elements in array 2
Array elements for array2
Output Format
Display the merged array

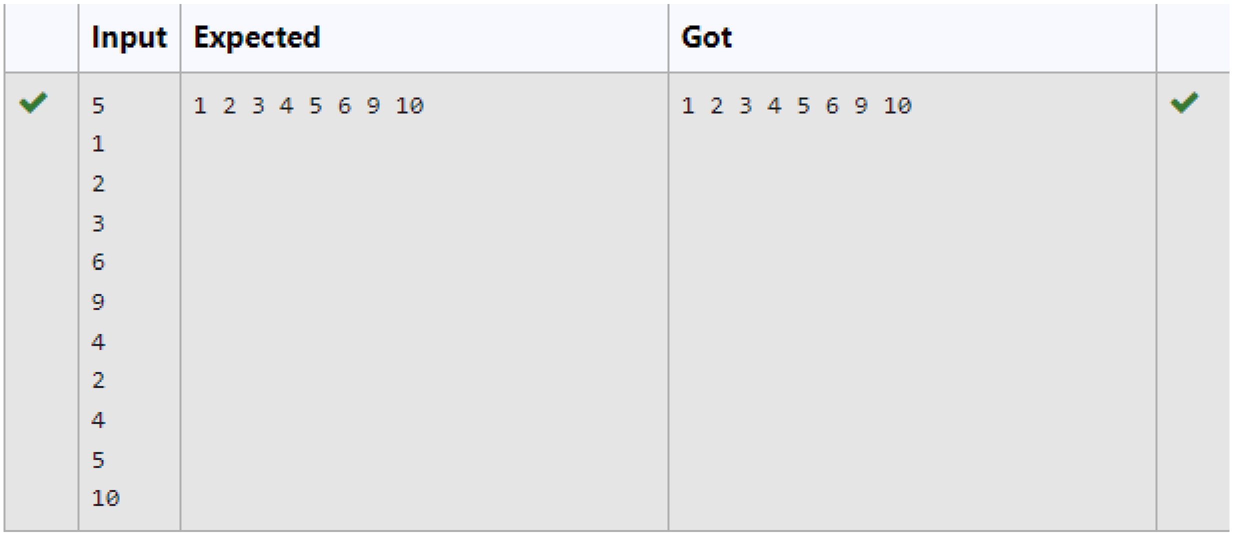
Sample Input 1

```
5
1
2
3
6
9
4
2
4
5
10
```

Sample Output 1

```
1 2 3 4 5 6 9 10
```

```
n = int(input())

arr = [int(input()) for _ in range(n)]

distinct_elements = set(arr)

print(*distinct_elements)
```



	Input	Expected	Got	
✓	5 1 2 3 6 9 4 2 4 5 10	1 2 3 4 5 6 9 10	1 2 3 4 5 6 9 10	✓

Ex. No. : 6.9 **Date: 4/05/2024**

Register No.: 231401001 **Name: Aafrin Fathima N**

Print Element Location

Write a program to print all the locations at which a particular element (taken as input) is found in a list and also print the total number of times it occurs in the list. The location starts from 1.

For example, if there are 4 elements in the array:
5
6
5
7

If the element to search is 5 then the output will be:
5 is present at location 1
5 is present at location 3
5 is present 2 times in the array.
Sample Test Cases

Test Case 1
Input

4
5
6
5
7
5

Output

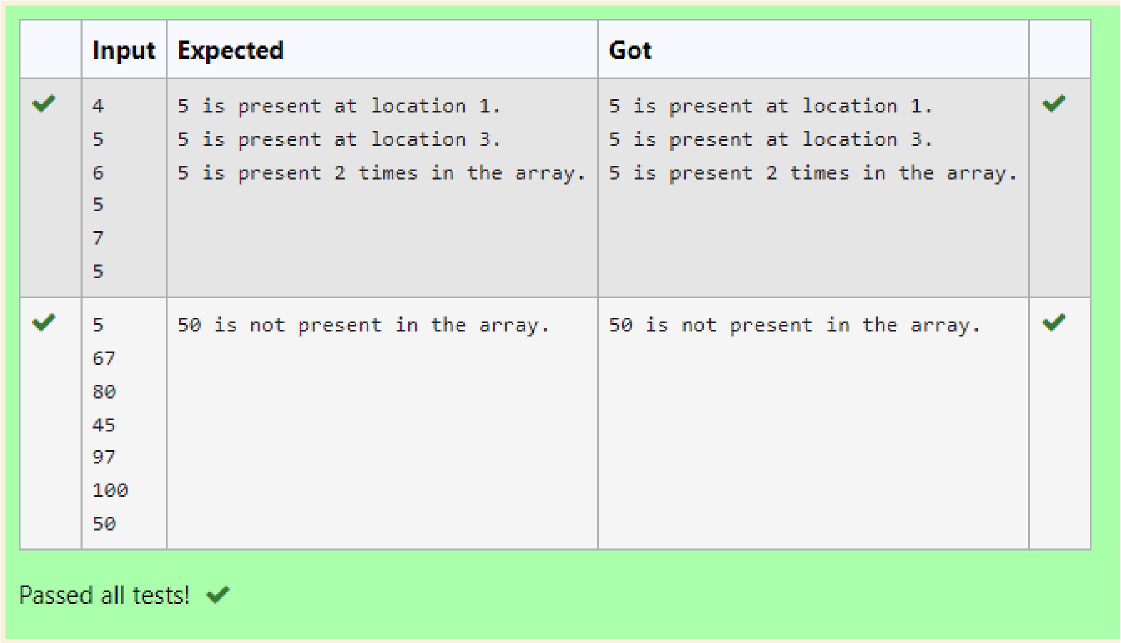
5 is present at location 1.
5 is present at location 3.
5 is present 2 times in the array.

Test Case 2
Input
5
67
80

45
97
100
50

Output
50 is not present in the array.

```
n=int(input())

arr = [int(input()) for _ in range(n)]

x=int(input())

count=0

for i in range(n):

    if arr[i]==x:

        print(x,"is present at location",i+1,end='.\n')

        count+=1

if count==0:

    print(x,"is not present in the array.")

else:

    print(x,"is present",count,"times in the array.")
```



	Input	Expected	Got	
✓	4 5 6 5 7 5	5 is present at location 1. 5 is present at location 3. 5 is present 2 times in the array.	5 is present at location 1. 5 is present at location 3. 5 is present 2 times in the array.	✓
✓	5 67 80 45 97 100 50	50 is not present in the array.	50 is not present in the array.	✓

Passed all tests! ✓

Ex. No. : 6.10 **Date: 4/05/2024**

Register No.: 231401001 **Name: Aafrin Fathima N**

Strictly increasing

Write a Python program to check if a given list is strictly increasing or not. Moreover, If removing only one element from the list results in a strictly increasing list, we still consider the list true
Input:
n : Number of elements
List1: List of values
Output
Print "True" if list is strictly increasing or decreasing else print "False"
Sample Test Case

Input

7

1

2

3

0

4

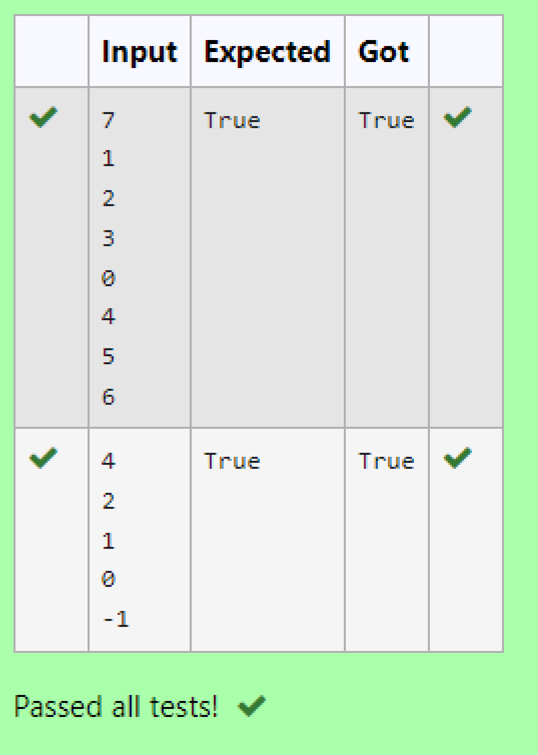
5

6

Output

True

```
def check(n, lst):
    def is_inc(l):
        return all(l[i] < l[i+1] for i in range(len(l) - 1))
    def is_dec(l):
        return all(l[i] > l[i+1] for i in range(len(l) - 1))
    if is_inc(lst) or is_dec(lst):
        return True
    for i in range(n):
        temp_lst = lst[:i] + lst[i+1:]
        if is_inc(temp_lst) or is_dec(temp_lst):
            return True
    return False
n = int(input())
lst = [int(input()) for _ in range(n)]
print(check(n, lst))
```



	Input	Expected	Got	
✓	7 1 2 3 0 4 5 6	True	True	✓
✓	4 2 1 0 -1	True	True	✓

Passed all tests! ✓